



UNITED STATES PATENT AND TRADEMARK OFFICE

mn
UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/725,016	12/02/2003	Serguei M. Belousov	2230.0020000/MBR/GSB	3183
54089 7590 06/04/2007 BARDMESSER LAW GROUP, P.C. 910 17TH STREET, N.W. SUITE 800 WASHINGTON, DC 20006			EXAMINER NGUYEN, PHILLIP H	
			ART UNIT 2191	PAPER NUMBER
			MAIL DATE 06/04/2007	DELIVERY MODE PAPER

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

Office Action Summary

Application No.

10/725,016

Applicant(s)

BELOUSSOV ET AL.

Examiner

Phillip H. Nguyen

Art Unit

2191

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

- 1) ☒ Responsive to communication(s) filed on 05 March 2007.
- 2a) ☐ This action is **FINAL**. 2b) ☒ This action is non-final.
- 3) ☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

- 4) ☒ Claim(s) 1-73 is/are pending in the application.
- 4a) Of the above claim(s) 58 and 65 is/are withdrawn from consideration.
- 5) ☐ Claim(s) _____ is/are allowed.
- 6) ☒ Claim(s) 1-57, 59-64 and 66-73 is/are rejected.
- 7) ☐ Claim(s) _____ is/are objected to.
- 8) ☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

- 9) ☐ The specification is objected to by the Examiner.
- 10) ☐ The drawing(s) filed on _____ is/are: a) ☐ accepted or b) ☐ objected to by the Examiner.
Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).
Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).
- 11) ☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

- 12) ☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).
- a) ☐ All b) ☐ Some * c) ☐ None of:
- ☐ Certified copies of the priority documents have been received.
 - ☐ Certified copies of the priority documents have been received in Application No. _____.
 - ☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

- 1) ☒ Notice of References Cited (PTO-892)
- 2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
- 3) ☐ Information Disclosure Statement(s) (PTO/SB/08)
Paper No(s)/Mail Date _____
- 4) ☐ Interview Summary (PTO-413)
Paper No(s)/Mail Date _____
- 5) ☐ Notice of Informal Patent Application
- 6) ☐ Other: _____

DETAILED ACTION

1. This action is in response to the amendment filed on 3/5/2007.
2. Per Applicant's request:
 - Claims 1, 27, 40, 53 and 67-71 have been amended.
 - Claims 1-57, 59-64 and 66-73 remain pending and have been considered below.

Response to Arguments

3. Applicant's arguments with respect to claims 1-73 have been considered but are moot in view of the new ground(s) of rejection.

Examiner Note:

4. Applicant appears to invoke 35 U.S.C. 112 6th paragraph in claims 53, 61-63, 65, 67, 68 by using "means-plus-function" language. However, Examiner notes that the claims recite sufficient structure, which is "computer program code" for performing those recited functions. While the claims pass the first of the three-prong test used to determine invocation of paragraph 6th, since they also recite sufficient structure within the claims to perform entirely recited functions, the claims are not in means-plus-function format, even if the claims use the term "means". Therefore, 35 U.S.C. 112 6th paragraph has not been invoked when considered these claims below.

Specification

5. The abstract of the disclosure is objected to because it contains more 150 words.

Correction is required. See MPEP § 608.01(b).

Claim Objections

6. Claim 59 objected to because of the following informalities: Since claim 58 has been canceled, claim 59 should be depending on claim 53. Appropriate correction is required.

Claim Rejections - 35 USC § 101

7. 35 U.S.C. 101 reads as follows:

Whoever invents or discovers any new and useful process, machine, manufacture, or composition of matter, or any new and useful improvement thereof, may obtain a patent therefor, subject to the conditions and requirements of this title.

8. Claims 67-68 are rejected under 35 U.S.C. 101 because the claimed invention is directed to non-statutory subject matter.

- Regarding claims 67 and 68 recite "a computer useable medium", which is disclosed as signals. The specification provides intrinsic evidence that the computer useable medium is intended to cover signals (see paragraph 82), such are currently not believed to enable the computer useable medium to act as a computer hardware component and realize its functionality absent being claimed in combination with the necessary hardware to receive and convert the signals to computer program logic.

Claim Rejections - 35 USC § 102

9. The following is a quotation of the appropriate paragraphs of 35 U.S.C. 102 that form the basis for the rejections under this section made in this Office action:

A person shall be entitled to a patent unless –

(e) the invention was described in (1) an application for patent, published under section 122(b), by another filed in the United States before the invention by the applicant for patent or (2) a patent granted on an application for patent by another filed in the United States before the invention by the applicant for patent, except that an international application filed under the treaty defined in section 351(a) shall have the effects for purposes of this subsection of an application filed in the United States only if the international application designated the United States and was published under Article 21(2) of such treaty in the English language.

10. Claims 1-7, 9, 10, 13-29, 31, 32, 35-45, 48-55, 57, 58, 61-73 are rejected under 35 U.S.C. 102(e) as being anticipated by Duesterwald et al. (United States Patent Application Publication No.: US 2003/0101330 A1).

As per claims 1, 27, 53, 67, 69 and 70:

Duesterwald discloses:

- identifying original instructions to be changed while the original instructions are being executed on a processor (see at least paragraph 54 **“intercept the various application instructions that are to be executed”**, also see FIGS. 4-5);
- copying the original instructions to a storage location (see at least paragraph 56 **“fragment is copied to on or more instruction buffer”**, also see FIGS. 4-5);
- adding a jump instruction to the copied instructions to return to a next instruction after the original instruction (see paragraph 54 **“DELI 100 is...injected into the**

application 102...so as to gain control over its execution", also see FIGS. 4-5); and

- **replacing the original instructions while the original instructions are in the process of being executed on the processor with mark instructions and a transfer of control to a hook (see at least paragraph 58 "application instructions are replaced with the patch code that is provided in the associated patch descriptor...", also see FIGS. 4-5), see at least paragraph 55 "DELI 100 jumps back to the application code");**

- o **wherein the original instructions are part of the instruction set of the processor available to a user (see at least paragraph 18 "code fragment which correspond to the instruction set of the hardware 104"); and**
- o **wherein a number of times the mark instructions have been executed is countable (see at least paragraph 48 "code fragments are executed by the application 102 under the control of the DELI 100...DELI 100 can therefore determine which code fragments are used most frequency... can make the determination of which pieces of code are hot").**

As per claim 2:

Duesterwald discloses:

Art Unit: 2191

- prior to the copying step, allowing a write operation on a page in memory wherein the original code is located (**It is inherent in Duesterwald's approach in order to copy the original instruction into a buffer**).

As per claim 3:

Duesterwald discloses:

- prior to the allocating step, masking interrupts (see at least paragraph 45 "**DELI 100 is injected into the application 102 with the injector 126...gains control over the application and its execution**" – This can consider masking the interrupt of the application).

As per claim 5:

Duesterwald discloses:

- after the replacing step, unmasking interrupt (see at least paragraph 49 "**DELI 100 jumps back to the application code and the execution of that code is resumed**" – This can consider as unmasking the interrupt of the application after replacing original code with patched code).

As per claims 6, 28 and 54:

Duesterwald discloses:

- wherein the original instructions are changed in reverse order (see at least paragraph 51 "**DELI 100 is to merely optimize the application**

Art Unit: 2191

execution...comprise rearranging and/or reconfiguring the code for better performance").

As per claims 7, 29 and 55:

Duesterwald discloses:

- wherein the mark instructions are the same length, in bytes, as the instructions to be changed (see at least paragraph 53 "**...dynamically replace them with new code fragments that do not require that functionality**" – This indicates that the length of patch instruction and the instructions to be changed are the same in order to perform the faulty or missing hardware functionality) .

As per claims 9, 31 and 57:

Duesterwald discloses:

- wherein the modified instructions include a resolver to determine a number of the instructions at a location of the original code that had already been executed (see at least paragraph 48 "**DELI 100 sees each piece of code that is executed. Through the monitoring process, the DELI 100 can, therefore, determine which code fragments are used more frequently**").

As per claims 10 and 32:

Duesterwald discloses:

Art Unit: 2191

- wherein the resolver determines a number of instructions that had already been executed using the mark instructions (see at least paragraph 49 “**determination of whether the code has been cached can be made with reference to, as noted above, identifiers (e.g., tags)...**”).

As per claims 13, 35 and 61:

Duesterwald discloses:

- enabling functionality of the copied instructions at the storage location (see at least paragraph 49 “**execution of the cached code...**”).

As per claims 14 and 62:

Duesterwald discloses:

- the enabling step comprises reconciling addressing in the instructions in the storage location (see at least paragraph 24 “**the original application’s text segment is still loaded at the same virtual address that it would normally have**”).

As per claim 15:

Duesterwald discloses :

- wherein the enabling step comprises alignment of instructions in the instructions at the storage location (see at least paragraph 51 “**...rearranging and/or reconfiguring the code for better performance**”).

Art Unit: 2191

As per claims 16, 36 and 63:

Duesterwald discloses:

- verifying that the original code is susceptible to patching (see at least paragraph 53 **"determine which call upon faulty or missing hardware functionality"**).

As per claims 17, 37 and 64:

Duesterwald discloses:

- wherein the verifying step determines whether any mark instructions are already present in the original instructions (see at least paragraph 53 **"the new code fragments can be cached such that, next time the original code fragments (i.e., a particular function) are required, the new code fragments can be executed with the cached..."** – a determination must be made in order to know the patched code (marked code) is already exist in the original code).

As per claim 18:

Duesterwald discloses:

- wherein the verifying step determines whether any copy protect instructions are already present in the original instructions (**Copy protection must be allow in Duesterwald's approach in order copy the original instructions to storage area**).

Art Unit: 2191

As per claim 19:

Duesterwald discloses:

- wherein the verifying step determines whether the original instructions include a suitable jump point that can be modified to the transfer of control to the hook (see at least paragraph 24 **"adding a DELI text segment at the end, and the start symbol (i.e., the entry point that is called by crt0) changed to the DELI entry point"**).

As per claim 20:

Duesterwald discloses:

- wherein the verifying step determines whether the original instructions represent valid instructions **(the original instructions must be valid instructions in order to perform dynamic patching)**.

As per claims 21 and 38:

Duesterwald discloses:

- placing the hook in the memory (see at least paragraph 52 **"several hooks that can be identified to the DELI 100 to permit code fragment replacement"**).

As per claim 22:

Duesterwald discloses:

Art Unit: 2191

- the hook has been previously placed in memory (see at least paragraph 52 **"several hooks that can be identified to the DELI 100 to permit code fragment replacement"**).

As per claims 23, 39 and 66:

Duesterwald discloses:

- wherein the replacing step use an atomic write to replace the original instructions **(It is inherent since there is no changed in instruction length)**.

As per claim 24:

Duesterwald discloses:

- wherein the atomic write replaces one instruction at a time (see at least paragraph 53 **"DELI controls very small portions of code such as code fragments and even individual instructions"**).

As per claim 25:

Duesterwald discloses:

- wherein the atomic write replaces multiple instructions at a time (see at least paragraph 53 **"DELI controls very small portions of code such as code fragments and even individual instructions"**).

As per claim 26:

Duesterwald discloses:

- wherein, for Intel IA-32 architecture, the atomic write uses any of "xchg," "lock cmpxchg8b," "lock cmpxchg," and "lock xchg" instructions (see at least paragraph 57 **"an X86 microprocessor"** – **any one of the instructions cited can be used in Duesterwald's approach**).

As per claims 40, 68 ad 71:

Duesterwald discloses:

- identifying original instructions to be changed while the original instructions are being executed on a processor (see at least paragraph 54 **"intercept the various application instructions that are to be executed"**, also see FIGS. 4-5);
- allocating a storage location for storing a functionality equivalent copy of the original instructions (see at least paragraph 21 **"code cache 124"**)
- copying the original instructions to the storage location (see at least paragraph 56 **"fragment is copied to on or more instruction buffer"**, also see FIGS. 4-5);
and
- replacing the original instructions while the original instructions are in the process of being executed on the processor with mark instructions and a transfer of control to a hook (see at least paragraph 58 **"application instructions are replaced with the patch code that is provided in the associated patch"**

descriptor...", also see FIGS. 4-5), see at least paragraph 55 "**DELI 100 jumps back to the application code**";

- wherein the original instructions are part of the instruction set of the processor available to user (see at least paragraph 18 "**code fragment which correspond to the instruction set of the hardware 104**"); and
- wherein a number of times the mark instructions have been executed is countable (see at least paragraph 48 "**code fragments are executed by the application 102 under the control of the DELI 100...DELI 100 can therefore determine which code fragments are used most frequency... can make the determination of which pieces of code are hot**").

As per claim 41:

Duesterwald discloses:

- prior to the copying step, allowing a write operation on a page in memory wherein the original code is located (**It is inherent in Duesterwald's approach in order to copy the original instruction into a buffer**).

As per claim 42:

Duesterwald discloses:

- adding a jump instruction to the copied instructions to return to a next instruction after the original instructions (see paragraph 54 "**DELI 100 is...injected into the**

application 102...so as to gain control over its execution", also see FIGS. 4-5).

As per claim 43:

Duesterwald discloses:

- wherein the original instructions are changed in reverse order (see at least paragraph 51 **"DELI 100 is to merely optimize the application execution...comprise rearranging and/or reconfiguring the code for better performance"**).

As per claim 44:

Duesterwald discloses:

- wherein the modified instructions include a resolver to determine a number of the instructions at a location of the original code that had already been executed (see at least paragraph 48 **"DELI 100 sees each piece of code that is executed. Through the monitoring process, the DELI 100 can, therefore, determine which code fragments are used more frequently"**).

As per claim 45:

Duesterwald discloses:

- wherein the resolver determines a number of instructions that had already been executed using the mark instructions (see at least paragraph 49 **"determination**

of whether the code has been cached can be made with reference to, as noted above, identifiers (e.g., tags)...”).

As per claim 48:

Duesterwald discloses:

- verifying that the original code is susceptible to patching (see at least paragraph 53 **“determine which call upon faulty or missing hardware functionality”**).

As per claim 49:

Duesterwald discloses:

- wherein the verifying step determines whether any mark instructions are already present in the original instructions (see at least paragraph 53 **“the new code fragments can be cached such that, next time the original code fragments (i.e., a particular function) are required, the new code fragments can be executed with the cached...”** – a determination must be made in order to know the patched code (marked code) is already exist in the original code).

As per claim 50:

Duesterwald discloses:

- wherein the verifying step determines whether any copy protect instructions are already present in the original instructions (**Copy protection must be allow in**

Art Unit: 2191

Duesterwald's approach in order copy the original instructions to storage area).

As per claim 51:

Duesterwald discloses:

- wherein the replacing step use an atomic write to replace the original instructions
(It is inherent since there is no changed in instruction length).

As per claim 52:

Duesterwald discloses:

- enabling functionality of the copied instructions at the storage location (see at least paragraph 49 "**execution of the cached code...**").

As per claims 72 and 73:

Duesterwald discloses:

- wherein the process of execution of the original instructions is not interrupted throughout the patching process (see at least paragraph 14 "**dynamically patching code, i.e. patching program code while the program is running**").

Claim Rejections - 35 USC § 103

1. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

(a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negated by the manner in which the invention was made.

2. Claims 8, 11, 12, 30, 33, 34, 46, 47, 56, 59 and 60 are rejected under 35 U.S.C. 103(a) as being unpatentable over Duesterwald et al (United States Patent Application Publication No.: US 20030101330 A1).

As per claims 8, 30 and 56:

Duesterwald does not explicitly disclose:

- wherein the mark instructions are shorter in length, in bytes, as the instructions to be changed, and include NOP (no operation) filler.

However, Applicant discloses that in the Intel X86 architecture in the specification, when an instruction is changed, its length is never increased (Paragraph 49). It would have been obvious to one having ordinary skill in the art at the time the invention was made to recognize that in Intel X86 architecture, any changed command is either the same length as the original instruction or is shorter with corresponding NOP instructions (no operation) in the remaining bytes.

Therefore, one would have been motivated to use this feature in the Intel X86 architecture for checking to see if the mark instructions are shorter in length, in bytes, as the instructions to be changed or any other useful reason for the invention.

As per claims 11, 33, 46 and 59:

Duesterwald does not explicitly disclose:

- if the number of instructions that had already been executed is less than a number of original instructions to be changed, the resolver calls the copied instructions at the storage location so as to imitate a "no patch installed" scenario.

However, it would have been obvious to one having ordinary skill in the art at the time of the invention was made to recognize that after the instructions had already been executed, the contents of the instructions had already been changed. So, if there is an interrupt occurs during the execution process, the hook is going to execute before the unchanged instructions of the original code not the changed instructions. Otherwise the patching result would be different than expected.

Therefore, one would have been motivated to add this step in the patching process to ensure the patching results are correctly.

As per claims 12, 34, 47 and 60:

Duesterwald discloses:

Art Unit: 2191

- wherein, after execution of the instructions at the storage location, the resolver returns control to the next instruction (see at least paragraph 49 “**DELI 100 jumps back to the application code and the execution of that code is resumed**”).

3. Claim 4 is rejected under 35 U.S.C. 103(a) as being unpatentable over Duesterwald et al (United States Patent Application Publication No.: US 20030101330 A1), in view of Scott et al. (United States Patent No.: US 6,615,329).

As per claim 4:

Duesterwald does not explicitly disclose:

- after the replacing step, disallowing (disabling) a write operation on the page in memory where the block of code is located.

However, Scott discloses an analogous method that disable (disallow) a write operation on the page in memory where the block of code is located to protect the area from unauthorized user (“**disable write operations to the protected area**” Col 9,53-54).

Therefore, It would have been obvious to one having ordinary skill in the art at the time the invention was made to modify Duesterwald's approach to allow disable write operation. One of ordinary skill would have been motivated to consider protecting the memory area by disable or disallow a write operation after data have been copied from memory to storage location.

Conclusion

4. The prior art made of record and not relied upon is considered pertinent to applicant's disclosure.

- Duesterwald et al. (United States Patent No.: US 6,915,513 B2) discloses a system and method for dynamically replace code.
- Duesterwald et al. (United States Patent No.: US 6,928,536 B2) discloses a dynamically execution layer interface for replacing instructions requiring unavailable hardware functionality with patch code and caching.

Any inquiry concerning this communication or earlier communications from the examiner should be directed to Phillip H. Nguyen whose telephone number is (571) 270-1070. The examiner can normally be reached on Monday - Thursday 10:00 AM - 3:00 PM EST.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Wei Y. Zhen can be reached on (571) 272-3708. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Art Unit: 2191

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see <http://pair-direct.uspto.gov>. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free). If you would like assistance from a USPTO Customer Service Representative or access to the automated information system, call 800-786-9199 (IN USA OR CANADA) or 571-272-1000.

P.N.
5/18/2007


WEI ZHEN
SUPERVISORY PATENT EXAMINER